

#6 : MIPS Programming II

Computer Architecture 2019/2020

João Soares & Ricardo Rocha

Computer Science Department, Faculty of Sciences, University of Porto

Program Structure

```
    .data                # data segment (constants and global variables)
_b1:  .byte 1           # byte (8 bits) with value 1
_h1:  .half 10          # half word (16 bits) with value 10
_w1:  .word 100         # word (32 bits) with value 100
_a1:  .byte 1, 2, 3, 4  # array of 4 bytes with values 1, 2, 3 and 4
_a2:  .word 0:100       # array of 100 words with values 0
_s1:  .ascii "abc\n"    # string not null terminated
_s2:  .asciiz "123"     # string null terminated
_e1:  .space 100        # leave 100 bytes of space

    .text               # text segment (program instructions)
_main:                  # main procedure
    ...
    li $v0, 10          # load code 10 for system call exit()
    syscall             # exit()
```

System Calls

To request a service, load the system call code into register `$v0` and arguments into registers `$a0–$a3` or `$f12` (floating point values).

Return values are put in register `$v0` or `$f0` (floating-point results).

Service	System call code	Arguments	Result
<code>print_int</code>	1	<code>\$a0 = integer</code>	
<code>print_float</code>	2	<code>\$f12 = float</code>	
<code>print_double</code>	3	<code>\$f12 = double</code>	
<code>print_string</code>	4	<code>\$a0 = string</code>	
<code>read_int</code>	5		integer (in <code>\$v0</code>)
<code>read_float</code>	6		float (in <code>\$f0</code>)
<code>read_double</code>	7		double (in <code>\$f0</code>)
<code>read_string</code>	8	<code>\$a0 = buffer, \$a1 = length</code>	
<code>sbrk</code>	9	<code>\$a0 = amount</code>	address (in <code>\$v0</code>)
<code>exit</code>	10		
<code>print_char</code>	11	<code>\$a0 = char</code>	
<code>read_char</code>	12		char (in <code>\$v0</code>)
<code>open</code>	13	<code>\$a0 = filename (string), \$a1 = flags, \$a2 = mode</code>	file descriptor (in <code>\$a0</code>)
<code>read</code>	14	<code>\$a0 = file descriptor, \$a1 = buffer, \$a2 = length</code>	num chars read (in <code>\$a0</code>)
<code>write</code>	15	<code>\$a0 = file descriptor, \$a1 = buffer, \$a2 = length</code>	num chars written (in <code>\$a0</code>)
<code>close</code>	16	<code>\$a0 = file descriptor</code>	
<code>exit2</code>	17	<code>\$a0 = result</code>	

MARS – MIPS Simulator

Main functionalities:

- Edit programs (**assembly**)
- Compile (**assembler**)
- Run and/or execute step by step
- See the memory contents and the values in the set of registers

Download Mars4_5.jar:

- <http://www.softpedia.com>

Command to execute:

- `java -jar Mars4_5.jar`

MARS – MIPS Simulator

The screenshot shows the MARS 4.5 MIPS simulator interface. The window title is "MARS 4.5". The menu bar includes "File", "Edit", "Run", "Settings", "Tools", and "Help". The toolbar contains icons for file operations and execution. A "Code editor" callout points to the main workspace. A "CPU registers" callout points to a table of registers. A "Compiler output window and processor input/output console" callout points to the bottom section of the interface.

Code editor

Run speed at max (no interaction)

Name	Number	Value
\$zero	0	0x00000000
\$at	1	0x00000000
\$v0	2	0x00000000
\$v1	3	0x00000000
\$a0	4	0x00000000
\$a1	5	0x00000000
\$a2	6	0x00000000
\$a3	7	0x00000000
\$t0	8	0x00000000
\$t1	9	0x00000000
\$t2	10	0x00000000
\$t3	11	0x00000000
\$t4	12	0x00000000
\$t5	13	0x00000000
\$t6	14	0x00000000
\$t7	15	0x00000000
\$s0	16	0x00000000
\$s1	17	0x00000000
\$s2	18	0x00000000
\$s3	19	0x00000000
\$s4	20	0x00000000
\$s5	21	0x00000000
\$s6	22	0x00000000
\$s7	23	0x00000000
\$t8	24	0x00000000
\$t9	25	0x00000000
\$k0	26	0x00000000
\$k1	27	0x00000000
\$gp	28	0x10008000
\$sp	29	0x7ffffeff
\$fp	30	0x00000000
\$ra	31	0x00000000
\$c		0x00400000
\$l		0x00000000
\$u		0x00000000

Compiler output window and processor input/output console

**CPU registers
<Name, Number, Value>**

MARS – MIPS Simulator

The screenshot shows the MARS MIPS simulator interface. The main window is divided into several sections. At the top, there is a menu bar (File, Edit, Run, Settings, Tools, Help) and a toolbar. Below the toolbar, there are two large yellow grid areas: the top one is labeled 'Text Segment' and the bottom one is labeled 'Data Segment'. Both are circled in red. A blue callout box points to the 'Text Segment' area with the text 'Text segment preview (binary code and source code)'. Another blue callout box points to the 'Data Segment' area with the text 'Data segment preview (memory contents)'. To the right of these segments is a register window titled 'Coproc 0' with a table of registers and their values. At the bottom, there is a 'Mars Messages' window with a 'Clear' button.

File Edit Run Settings Tools Help

Text Segment

Data Segment

0x10010000 (data) Hexadecimal Addresses

Mars Messages

Clear

Name	Number	Value
\$zero	0	0x00000000
\$at	1	0x00000000
\$v0	2	0x00000000
\$v1	3	0x00000000
\$a0	4	0x00000000
\$a1	5	0x00000000
\$a2	6	0x00000000
\$a3	7	0x00000000
\$t0	8	0x00000000
\$t1	9	0x00000000
\$t2	10	0x00000000
\$t3	11	0x00000000
\$t4	12	0x00000000
\$t5	13	0x00000000
\$t6	14	0x00000000
\$t7	15	0x00000000
\$s0	16	0x00000000
\$s1	17	0x00000000
\$s2	18	0x00000000
\$s3	19	0x00000000
\$s4	20	0x00000000
	21	0x00000000
	22	0x00000000
	23	0x00000000
	24	0x00000000
	25	0x00000000
	26	0x00000000
	27	0x00000000
	28	0x10008000
\$sp	29	0x7ffffc
\$fp	30	0x00000000
\$ra	31	0x00000000
pc		0x00400000
hi		0x00000000
lo		0x00000000

MARS – MIPS Simulator

The screenshot shows the MARS MIPS simulator interface. The main window displays assembly code on a yellow grid background. A blue callout bubble points to the 'Run' menu item, with the text 'Compile (Run -> Assemble)'. Another blue callout bubble points to the 'File' menu, with the text 'Create new program (File -> New), edit and save (File -> Save)'. The right side of the window shows a 'Registers' panel with a table of register names, numbers, and values. The status bar at the bottom left shows 'Line: 13 Column: 9' and a 'Show Line Numbers' button. A 'Clear' button is visible in the bottom left corner of the main window.

File Edit Run Settings Tools Help

Run speed at max (no interaction)

Execute

asm

2 li \$t1, 0
3 lw \$t2, _X
5 add \$t1, \$t1, \$t2
6 sw \$t1, _X
7
8 addiu \$v0, \$zero, 10
9 syscall
10
11
12
13

Line: 13 Column: 9 Show Line Numbers

Clear

Name	Number	Value
\$zero	0	0x00000000
\$at	1	0x00000000
\$v0	2	0x00000000
\$v1	3	0x00000000
\$a0	4	0x00000000
\$a1	5	0x00000000
\$a2	6	0x00000000
\$a3	7	0x00000000
\$t0	8	0x00000000
\$t1	9	0x00000000
\$t2	10	0x00000000
\$t3	11	0x00000000
\$t4	12	0x00000000
\$t5	13	0x00000000
\$t6	14	0x00000000
\$t7	15	0x00000000
\$s0	16	0x00000000
\$s1	17	0x00000000
\$s2	18	0x00000000
\$s3	19	0x00000000
\$s4	20	0x00000000
\$s5	21	0x00000000
\$s6	22	0x00000000
\$s7	23	0x00000000
\$t8	24	0x00000000
\$t9	25	0x00000000
\$k0	26	0x00000000
\$k1	27	0x00000000
\$gp	28	0x10008000
\$sp	29	0x7ffffc
\$fp	30	0x00000000
\$ra	31	0x00000000
pc		0x00400000
hi		0x00000000
lo		0x00000000

MARS – MIPS Simulator

The screenshot shows the MARS MIPS simulator interface. At the top, there is a menu bar with 'File', 'Edit', 'Run', 'Settings', 'Tools', and 'Help'. Below the menu bar, the status bar indicates 'Run speed at max (no interaction)'. A blue callout box points to the 'Run' menu, containing the text: 'Run program (Run -> Go) ou execute step by step (Run -> Step)'. The main window is divided into several panes. The top pane, titled 'Text Segment', displays assembly code with columns for 'Bkpt', 'Address', 'Code', 'Basic', and 'Source'. A blue callout box points to this pane, containing the text: 'Lines <Address, Binary code, Source code>'. Below this is a pane showing memory addresses and their corresponding values, with a blue callout box pointing to it, containing the text: '1 word pairs <Address, Value>'. At the bottom, there is an assembly log pane showing the message: 'Assemble: operation completed successfully.' and a 'Clear' button. On the right side, there is a register window titled 'Coprocs' with tabs for 'Coproc 1' and 'Coproc 0'. It contains a table of registers with their names, numbers, and values.

Name	Number	Value
\$zero	0	0x00000000
\$at	1	0x00000000
\$v0	2	0x00000000
\$v1	3	0x00000000
\$a0	4	0x00000000
\$a1	5	0x00000000
\$a2	6	0x00000000
\$a3	7	0x00000000
\$t0	8	0x00000000
\$t1	9	0x00000000
\$t2	10	0x00000000
\$t3	11	0x00000000
\$t4	12	0x00000000
\$t5	13	0x00000000
\$t6	14	0x00000000
\$t7	15	0x00000000
\$s0	16	0x00000000
\$s1	17	0x00000000
\$s2	18	0x00000000
\$s3	19	0x00000000
\$s4	20	0x00000000
\$s5	21	0x00000000
\$s6	22	0x00000000
\$s7	23	0x00000000
\$t8	24	0x00000000
\$t9	25	0x00000000
\$k0	26	0x00000000
\$k1	27	0x00000000
\$gp	28	0x10008000
\$sp	29	0x7ffffeffc
\$fp	30	0x00000000
\$ra	31	0x00000000
pc		0x00400000
hi		0x00000000
lo		0x00000000

MARS – MIPS Simulator

